Distance Transform Computation for Digital Distance Functions

Robin Strand, Nicolas Normand

Distance Transform Computation for Digital Distance Functions

Robin Strand[a], Nicolas Normand[b]

[a] *Centre for Image Analysis, Uppsala University, Box 337, SE-75105 Uppsala, Sweden*
[b] *LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597, Polytech Nantes, La Chantrerie, 44306 Nantes Cedex 3, France*

Abstract

In image processing, the distance transform (DT), in which each object grid point is assigned the distance to the closest background grid point, is a powerful and often used tool. In this paper, distance functions defined as minimal cost-paths are used and a number of algorithms that can be used to compute the DT are presented. We give proofs of the correctness of the algorithms.

Keywords: distance function, distance transform, weighted distances, neighborhood sequences

---

```
@article{strand2012tcs,
    Author = {Strand, Robin and Normand, Nicolas},
    Doi = {10.1016/j.tcs.2012.05.010},
    Journal = {Theoretical Computer Science},
    Pages = {80-93},
    Title = {Distance Transform Computation for Digital Distance Functions},
    Volume = {448},
    Year = {2012}}
```

The original publication is available at www.sciencedirect.com

SHERPA/RoMEO: green
`http://www.sherpa.ac.uk/romeo/search.php?jrule=ISSN&search=0304-3975`

Email addresses: `robin@cb.uu.se` (Robin Strand), `Nicolas.Normand@polytech.univ-nantes.fr` (Nicolas Normand)

Distance Transform Computation for Digital Distance Functions

Robin Strand[a], Nicolas Normand[b]

[a] *Centre for Image Analysis, Uppsala University, Box 337, SE-75105 Uppsala, Sweden*
[b] *LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597, Polytech Nantes, La Chantrerie, 44306 Nantes Cedex 3, France*

Abstract

In image processing, the distance transform (DT), in which each object grid point is assigned the distance to the closest background grid point, is a powerful and often used tool. In this paper, distance functions defined as minimal cost-paths are used and a number of algorithms that can be used to compute the DT are presented. We give proofs of the correctness of the algorithms.

Keywords: distance function, distance transform, weighted distances, neighborhood sequences

1. Introduction

In [1], an algorithm for computing distance transforms (DTs) using the basic city-block (horizontal and vertical steps are allowed) and chessboard (diagonal steps are allowed in conjunction with the horizontal and vertical steps) distance functions was presented in [1]. These distance functions are defined as shortest paths and the corresponding distance maps can be computed efficiently. Since these path-based distance functions are defined by the cost of discrete paths, we call them *digital* distance functions.

There are two commonly used generalizations of the city-block and chessboard distance functions, the *weighted distances* [2, 3, 4], and *distances based on neighborhood sequences* (ns-distances) [5, 6, 7, 8] . The weighted distance

Email addresses: `robin@cb.uu.se` (Robin Strand), `Nicolas.Normand@polytech.univ-nantes.fr` (Nicolas Normand)

is defined as the cost of a minimal cost-path and the ns-distance is defined as a shortest path in which the neighborhood that is allowed in each step is given by a neighborhood sequence. With weighted distances, a two-scan algorithm is sufficient for any point-lattice, see [2, 9]. For ns-distances, three scans are needed for computing correct DTs on a square grid [10].

In this paper, we consider the *weighted ns-distance* [11, 12, 13] in which both weights and a neighborhood sequence are used to define the distance function. By using "optimal" parameters (weights and neighborhood sequence), the asymptotic shape of the discs with this distance function is a twelve-sided polygon, see [11]. The relative error is thus asymptotically $(1/\cos(\pi/12)-1)/((1/\cos(\pi/12)+1)/2) \approx 3.5\%$ using only 3×3 neighborhoods when computing the DT. In other words, we have a close to exact approximation of the Euclidean distance still using the path-based approach with connectivities corresponding to small neighborhoods. Some different algorithms for computing the distance transform using the weighted ns-distance functions are given in this paper.

The paper is organized as follows: First, some basic notions are given and the definition of weighted ns-distances is given. In Section 3, algorithms using an additional DT holding the *length* of the paths that define the distance values are presented. The notion of distance propagating path is introduced to prove that correct DTs are computed. In Section 4, a look-up table that holds the value that should be propagated in each direction is used to compute the DT. The third approach considered here work for metric distance functions with periodic neighborhood sequences. A large mask that holds all distance information corresponding to the first period of the neighborhood sequence is used.

2. Weighted distances based on neighborhood sequences

The distance function considered here is defined by a neighborhood sequence using two neighborhoods and two weights. The neighborhoods are defined as follows

$$\mathcal{N}_1 = \{(\pm 1, 0), (0, \pm 1)\} \text{ and } \mathcal{N}_2 = \{(\pm 1, \pm 1)\}.$$

Two grid points $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{Z}^2$ are strict r-neighbors, $r \in \{1, 2\}$, if $\mathbf{p}_2 - \mathbf{p}_1 \in \mathcal{N}_r$. Neighbors of higher order can also be defined, but in this paper, we will use only 1- and 2-neighbors. Let

$$\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2.$$

The points $\mathbf{p}_1, \mathbf{p}_2$ are 2-neighbors (or *adjacent*) if $\mathbf{p}_2 - \mathbf{p}_1 \in \mathcal{N}$, i.e., if they are strict r-neighbors for some r. A ns B is a sequence $B = (b(i))_{i=1}^{\infty}$, where each $b(i)$ denotes a neighborhood relation in $\mathbb{Z}^2$. If B is periodic, i.e., if for some finite, strictly positive $l \in \mathbb{Z}_+$, $b(i) = b(i+l)$ is valid for all $i \in \mathbb{N}^*$, then we write $B = (b(1), b(2), \ldots, b(l))$.

The following notation is used for the number of 1:s and 2:s in the ns B up to position k.

$$\mathbf{1}_B^k = |\{i : b(i) = 1, 1 \le i \le k\}| \text{ and } \mathbf{2}_B^k = |\{i : b(i) = 2, 1 \le i \le k\}|.$$

A *path* in a grid, denoted $\mathcal{P}$, is a sequence $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n$ of adjacent grid points. A path is a *B-path* of length $\mathcal{L}(\mathcal{P}) = n$ if, for all $i \in \{1, 2, \ldots, n\}$, $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$ are $b(i)$-neighbors. The number of 1-steps and strict 2-steps in a given path $\mathcal{P}$ is denoted $\mathbf{1}_{\mathcal{P}}$ and $\mathbf{2}_{\mathcal{P}}$, respectively.

Definition 1. Given the ns B, the ns-distance $d(\mathbf{p}_0, \mathbf{p}_n; B)$ between the points $\mathbf{p}_0$ and $\mathbf{p}_n$ is the length of a shortest B-path between the points.

Let the real numbers α and β (the *weights*) and a B-path $\mathcal{P}$ of length n, where exactly l ($l \le n$) pairs of adjacent grid points in the path are strict 2-neighbors be given. The *cost of the (α,β)-weighted B-path $\mathcal{P}$* is $\mathcal{C}_{\alpha,\beta}(\mathcal{P}) = (n-l)\alpha + l\beta$. The B-path $\mathcal{P}$ between the points $\mathbf{p}_0$ and $\mathbf{p}_n$ is a *(α,β)-weighted minimal cost B-path between the points $\mathbf{p}_0$ and $\mathbf{p}_n$* if no other (α,β)-weighted B-path between the points has lower cost than the (α,β)-weighted B-path $\mathcal{P}$.

Definition 2. Given the ns B and the weights α, β, the weighted ns-distance $d_{\alpha,\beta}(\mathbf{p}_0, \mathbf{p}_n; B)$ is the cost of a (α,β)-weighted minimal cost B-path between the points.

The following theorem is from [11].

Theorem 1 (Weighted ns-distance in $\mathbb{Z}^2$)**.** *Let the ns B, the weights α, β s.t. $0 < \alpha \le \beta \le 2\alpha$, and the point $(x,y) \in \mathbb{Z}^2$, where $x \ge y \ge 0$, be given. The weighted ns-distance between $\mathbf{0}$ and (x,y) is given by*

$$\begin{aligned} d_{\alpha,\beta}(\mathbf{0}, (x,y); B) &= (2k - x - y) \cdot \alpha + (x + y - k) \cdot \beta \\ \textit{where } k &= \min_l : l \ge \max\left(x, x + y - \mathbf{2}_B^l\right). \end{aligned}$$

Note if $B = (1)$ then $k = x + y$ so $d(0, (x,y); (1)) = (x+y)\alpha$ which is α times the city-block distance whereas if $B = (2)$ then $k = x$ and $d(0, (x,y); (2)) = (x-y)\alpha + y\beta$ which is the (α,β)-weighted distance.

3. Computing the distance transform using path-length information

In this section, the computation of DTs using the distance function defined in the previous section will be considered. Since the size of a digital image when stored in a computer is finite, we define the *image domain* as a finite subset of $\mathbb{Z}^2$ denoted $\mathcal{I}$. In this paper we use image domains of the form

$$\mathcal{I} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \tag{1}$$

Definition 3. We call the function $F : \mathcal{I} \longrightarrow \mathbb{R}_0^+$ an *image*.

Note that real numbers are allowed in the range of F. We denote the *object* X and the *background* is $\overline{X} = \mathbb{Z}^2 \setminus X$. We denote the distance transform for path-based distances with $DT_{\mathcal{C}}$, where the subscript $\mathcal{C}$ indicates that costs of paths are computed.

Definition 4. The *distance transform* $DT_{\mathcal{C}}$ of an object $X \subset \mathcal{I}$ is the mapping

$$\begin{aligned} DT_{\mathcal{C}} : \mathcal{I} &\rightarrow \mathbb{R}_0^+ \text{ defined by} \\ \mathbf{p} &\mapsto d\left(\mathbf{p}, \overline{X}\right), \text{ where} \\ d\left(\mathbf{p}, \overline{X}\right) &= \min_{\mathbf{q} \in \overline{X}} \left\{ d\left(\mathbf{p}, \mathbf{q}\right) \right\}. \end{aligned}$$

For weighted ns-distances, the size of the neighborhood allowed in each step is determined by the *length* of the minimal cost-paths (not the cost). In the first approach to compute the DT, an additional transform, $DT_{\mathcal{L}}$ that holds the length of the minimal cost path at each point is used.

Definition 5. The set of transforms $\{DT_{\mathcal{L}}^i\}$ of an object $X \subset \mathbb{Z}^2$ is defined by all mappings $DT_{\mathcal{L}}^i$ that satisfy

$$\begin{aligned} DT_{\mathcal{L}}^i(\mathbf{p}) &= d_{1,1}\left(\mathbf{p}, \mathbf{q}; B\right), \text{ where} \\ \mathbf{q} \text{ is such that } d_{\alpha,\beta}\left(\mathbf{q}, \mathbf{p}; B\right) &= d_{\alpha,\beta}\left(\mathbf{p}, \overline{X}; B\right). \end{aligned}$$

See Figure 1 for an example showing $DT_{\mathcal{C}}$ and some different $DT_{\mathcal{L}}$ (superscipt omitted when it is not explicitly needed) of an object.

When $\alpha = \beta = 1$, $DT_{\mathcal{L}}$ is uniquely defined and $DT_{\mathcal{C}} = DT_{\mathcal{L}}$. Example 1 illustrates that $DT_{\mathcal{L}}$ is not always uniquely defined when $\alpha \neq \beta$. We will see that despite this, the correct distance values are propagated by natural extensions of well-known algorithms when $DT_{\mathcal{C}}$ is used together with $DT_{\mathcal{L}}^{i}$ for *any* i are used to propagate the distance values.

We now introduce the notion of distance propagating path.

0	0	0	0	0	0	0
0	0	α	α	α	0	0
0	α	β	2α	β	α	0
0	α	2α	3α	2α	α	0
0	α	β	2α	β	α	0
0	0	α	α	α	0	0
0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	1	2	1	1	0
0	1	2	3	2	1	0
0	1	1	2	1	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

(b)

Figure 1: Distance transforms for $B = (2, 1)$ and $\alpha \le \beta \le 2\alpha$. The background is shown in white, $DT_{\mathcal{C}}$ is shown in (a) and a $DT_{\mathcal{L}}$ is shown in (b).

Definition 6. Given an object grid point $\mathbf{p} \in X$, a minimal cost B-path $\mathcal{P}_{\mathbf{q},\mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n = \mathbf{p} \rangle$, where $\mathbf{q} \in \overline{X}$ is a background grid point, is a *distance propagating B-path* if

$$\begin{array}{ll} \texttt{(i)} & \mathcal{C}_{\alpha,\beta}\left(\langle \mathbf{p}_0, \ldots, \mathbf{p}_i \rangle\right) = DT_{\mathcal{C}}\left(\mathbf{p}_i\right) \text{ for all } i \text{ and} \\ \texttt{(ii)} & \mathbf{p}_i, \mathbf{p}_{i+1} \text{ are } b\left(DT_{\mathcal{L}}^{j}\left(\mathbf{p}_i\right) + 1\right) - \text{neighbors for all } i, \end{array}$$

for all j.

If property `(i)` in the definition above is fulfilled, then we say that $\mathcal{P}_{\mathbf{p},\mathbf{q}}$ *is represented by* $DT_{\mathcal{C}}$ and if property `(ii)` is fulfilled, then $\mathcal{P}_{\mathbf{p},\mathbf{q}}$ *is represented by* $DT_{\mathcal{L}}^{j}$. Note that when $\alpha = \beta$, then `(i)` implies `(ii)`.

If we can guarantee that there is such a path for every object grid point, then the distance transform can be constructed by locally propagating distance information from $\overline{X}$ to any $\mathbf{p} \in X$. Now, a number of definitions will be introduced. Using these definitions, we can show that there is always a distance propagating path when the weighted ns-distance function is used. The following definitions are illustrated in Example 1 and 2.

Definition 7. Let α, β such that $0 < \alpha \leq \beta \leq 2\alpha$, a ns B, an object X, and a point $\mathbf{p} \in X$ be given. A minimal cost B-path $\mathcal{P}_{\mathbf{q},\mathbf{p}}$, where $\mathbf{q} \in \overline{X}$, such that $d_{\alpha,\beta}(\mathbf{p}, \mathbf{q}; B) = d_{\alpha,\beta}\left(\mathbf{p}, \overline{X}; B\right)$ is a minimal cost B-path with *minimal number of* 2*-steps* if, for all paths $\mathcal{Q}_{\mathbf{q}',\mathbf{p}}$ with $\mathbf{q}' \in \overline{X}$ such that $\mathcal{C}_{\alpha,\beta}(\mathcal{Q}_{\mathbf{q}',\mathbf{p}}) = \mathcal{C}_{\alpha,\beta}(\mathcal{P}_{\mathbf{q},\mathbf{p}})$, we have

$$\mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}}} \leq \mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}}}.$$

In other words, if there are several paths defining the distance at a point $\mathbf{p}$, the path with the least number of 2-steps is a minimal cost B-path with *minimal number of* 2*-steps*. See Example 1 and 2.

Remark 1. A minimal cost-path with minimal number of 2-steps is a minimal cost-path of maximal length.

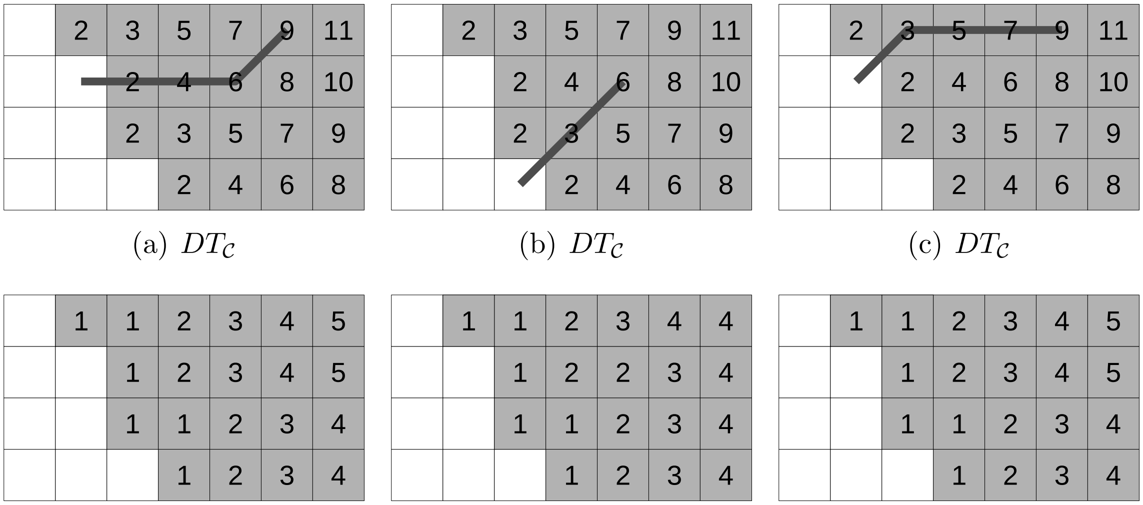
Definition 8. Let α, β such that $0 < \alpha \leq \beta \leq 2\alpha$, a ns B, and points $\mathbf{p}, \mathbf{q} \in \mathbb{Z}^2$ be given. The minimal cost (α, β)-weighted B-path $\mathcal{P}_{\mathbf{p},\mathbf{q}} = \langle \mathbf{p} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n = \mathbf{q} \rangle$ is a *fastest* minimal cost (α, β)-weighted B-path between $\mathbf{p}$ and $\mathbf{q}$ if there is an i, $0 \leq i \leq n$ such that

$$\mathbf{2}_{\langle \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_i \rangle} = \mathbf{2}_B^i \text{ and } \mathbf{2}_{\langle \mathbf{p}_{i+1}, \mathbf{p}_{i+2}, \ldots, \mathbf{p}_n \rangle} = 0.$$

In other words, the minimal cost path between two points in which the 2-steps occur after as few steps as possible is a *fastest* minimal cost (α, β)-weighted B-path. See Example 1 and 2.

Example 1. This example illustrates that a path that is not a fastest path is not necessarily represented by $DT_{\mathcal{L}}^i$ for some i. Consider the (part of the) object showed in Figure 2(a)–(f). The parameters $(\alpha, \beta) = (2, 3)$ and $B = (2, 2, 1)$ are used. In (d)–(f), some $DT_{\mathcal{L}}^i$:s are shown. The path in (a) has minimal number of 2-steps, but it is not a fastest path and is not represented by the $DT_{\mathcal{L}}$ in (e). The path in (b) has not minimal number of 2-steps. The path in (c) *is* a fastest path with minimal number of 2-steps and is also represented by all $DT_{\mathcal{L}}$ in (d)–(f). The paths shown in (b) and (c) are distance propagating paths, and the path in (a) is not a distance propagating path for $DT_{\mathcal{L}}$ in (e).

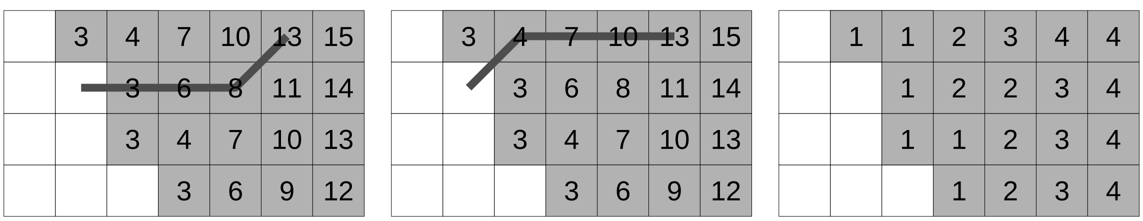
Example 2. In Figure 3(a)–(c), $B = (2, 2, 1)$ and $(\alpha, \beta) = (3, 4)$ are used. In (a), the $DT_{\mathcal{C}}$ of an object and a minimal cost (α, β)-weighted B-path with minimal number of 2-steps that is not distance propagating is shown.



(a) $DT_{\mathcal{C}}$ (b) $DT_{\mathcal{C}}$ (c) $DT_{\mathcal{C}}$

(d) $DT_{\mathcal{L}}$ (e) $DT_{\mathcal{L}}$ (f) $DT_{\mathcal{L}}$

Figure 2: Distance transform using $(\alpha, \beta) = (2, 3)$ and $B = (2, 2, 1)$ for a part of an object in $\mathbb{Z}^2$, see Example 1.



(a) $DT_{\mathcal{C}}$ (b) $DT_{\mathcal{C}}$ (c) $DT_{\mathcal{L}}$

Figure 3: Distance transform using $(\alpha, \beta) = (3, 4)$ and $B = (2, 2, 1)$ for a part of an object in $\mathbb{Z}^2$, see Example 2.

A distance propagating *fastest* minimal cost (α, β)-weighted B-path with minimal number of 2-steps is shown in (b). The $DT_{\mathcal{L}}$ that corresponds to $DT_{\mathcal{C}}$ in (a)–(b) is shown in (c).

The following theorem says that a path satisfying Definition 7 and 8 is a distance propagating path as defined in Definition 6. The theorem is proved in Lemma 2 and Lemma 3 below.

Theorem 2. *If the B-path $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ ($\mathbf{p} \in X$, $\mathbf{q} \in \overline{X}$ such that $d_{\alpha,\beta}\left(\mathbf{p}, \overline{X}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}}\right)$) is a fastest minimal cost B-path with minimal number of 2-steps then $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ is a distance propagating B-path.*

Intuitively, we want the path to be of maximal length (*B-path with minimal number of 2-steps, see Remark 1*) and among the paths with this property, the path such that the 2-steps appear after as few steps as possible (*fastest B-path*). The algorithms we present will always be able to propagate correct distance values along such paths.

Lemma 1 will be used in the proofs of Lemma 2 and Lemma 3. It is a direct consequence of Theorem 1. In Lemma 1, $B(k) = (b(i))_{i=k}^{\infty}$

Lemma 1. *Given α, β such that $0 < \alpha \leq \beta \leq 2\alpha$, the ns B, the points $\mathbf{p}, \mathbf{q}$, and an integer $k \geq 1$, we have*

$$d_{\alpha,\beta}(\mathbf{p}, \mathbf{q}; B) \leq d_{\alpha,\beta}(\mathbf{p}, \mathbf{q}; B(k)) + (2\alpha - \beta)(k-1).$$

We consider the case $\alpha < \beta$. Lemma 2 and Lemma 3 gives the proof of Theorem 2 for weighted ns-distances.

Lemma 2. *Let the weights α, β such that $0 < \alpha < \beta \leq 2\alpha$, the ns B, and the point $\mathbf{p} \in X$ be given. Any fastest minimal cost (α, β)-weighted B-path $\mathcal{P}_{\mathbf{q},\mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n = \mathbf{p} \rangle$ (for some $\mathbf{q} \in \overline{X}$ such that $d_{\alpha,\beta}\left(\mathbf{p}, \overline{X}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}}\right)$) satisfies* `(i)` *in Definition 6, i.e.,*

$$d_{\alpha,\beta}\left(\mathbf{p}_i, \overline{X}; B\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right) \ \forall i : 0 \leq i \leq n. \tag{2}$$

Proof. First we note that there always exists a $\mathbf{q} \in \overline{X}$ such that there is a fastest minimal cost (α, β)-weighted B-path $\mathcal{P}_{\mathbf{q},\mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n = \mathbf{p} \rangle$. To prove (2), assume that there is a $\mathbf{q}' \in \overline{X}$ and a path $\mathcal{Q}_{\mathbf{q}',\mathbf{p}} = \langle \mathbf{q}' = \mathbf{p}'_0, \mathbf{p}'_1, \ldots, \mathbf{p}'_k = \mathbf{p}_i, \mathbf{p}'_{k+1}, \ldots, \mathbf{p}'_m = \mathbf{p} \rangle$ for some i such that

$$\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right) > \mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right)$$

`Case i`: $\mathcal{L}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right) < \mathcal{L}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right)$

`Case i(a)`: $\mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}} > \mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_i}}$

Since $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ is a fastest path, $\mathbf{2}_{\mathcal{P}_{\mathbf{p}_i,\mathbf{p}}} = 0$. This implies that $\mathcal{P}_{\mathbf{p}_i,\mathbf{p}}$ is a minimal cost (α, β)-weighted B-path for $B = (1)$ and since, by Theorem 1, any ns generates distances less than (or equal to) $B = (1)$, $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{p}_i,\mathbf{p}}\right) \leq \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_i,\mathbf{p}}\right)$. Thus, $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right) + \mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{p}_i,\mathbf{p}}\right) < \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right) + \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_i,\mathbf{p}}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}}\right)$. This contradicts that $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ is a minimal cost-path, so this case can not occur.

`Case i(b)`: $\mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}} \leq \mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_i}}$

Since $\mathcal{L}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right) = \mathcal{L}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right) - L$ for some positive integer L, we have

$$\mathbf{1}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}} + \mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}} = \mathcal{L}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right) = \mathcal{L}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right) - L = \mathbf{1}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_i}} + \mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_i}} - L.$$

Using $\mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}} \leq \mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_i}}$ and $\alpha \leq \beta$, we get

$$\mathbf{1}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}}\alpha + \mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}}\beta \leq \mathbf{1}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_i}}\alpha + \mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_i}}\beta - L\alpha.$$

Thus, $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right) \leq \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right) - L\alpha$. By Lemma 1, $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{p}_i,\mathbf{p}}\right) \leq \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_i,\mathbf{p}}\right) + (2\alpha - \beta)\, L$.

We use these results and get $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right) + \mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{p}_i,\mathbf{p}}\right) \leq \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right) - L\alpha + \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_i,\mathbf{p}}\right) + (2\alpha - \beta)\, L = \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right) + \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_i,\mathbf{p}}\right) + (\alpha - \beta)\, L < \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}}\right)$.

This contradicts that $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ is a minimal cost-path.

`Case ii`: $\mathcal{L}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right) \geq \mathcal{L}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_i}\right)$

Now, $\mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}} < \mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_i}} \leq \mathbf{2}^i_B$.

Construct the path (*not* a B-path) $\mathcal{Q}'_{\mathbf{q}',\mathbf{p}} = \langle \mathbf{q}' = \mathbf{p}'_0, \mathbf{p}'_1, \ldots, \mathbf{p}'_k = \mathbf{p}_i, \mathbf{p}_{i+1}, \ldots, \mathbf{p}_n = \mathbf{p}\rangle$ of length $n' = \mathcal{L}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}\right) + \mathcal{L}\left(\mathcal{P}_{\mathbf{p}_i,\mathbf{p}_n}\right) \geq \mathcal{L}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_n}\right) = n$. We have $\mathbf{2}_{\mathcal{Q}'_{\mathbf{q}',\mathbf{p}_n}} = \mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_i}} + \mathbf{2}_{\mathcal{P}_{\mathbf{p}_i,\mathbf{p}_n}} < \mathbf{2}^i_B + \mathbf{2}^{n-i}_{B(i+1)} = \mathbf{2}^n_B \leq \mathbf{2}^{n'}_B$. This means that there is a B-path $\mathcal{Q}''_{\mathbf{q}',\mathbf{p}_n}$ (obtained by permutation of the positions of the 1-steps and 2-steps in $\mathcal{Q}'_{\mathbf{q}',\mathbf{p}_n}$) of length n' such that $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}''_{\mathbf{q}',\mathbf{p}_n}\right) < \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_n}\right)$, which contradicts that $\mathcal{P}_{\mathbf{q},\mathbf{p}_n}$ is a minimal cost path.

The assumption is false, since a contradiction follows for all cases. □

Left to prove of Theorem 2 is that there is a path fulfilling the previous lemma that is represented by $DT_{\mathcal{L}}$. This is necessary for the path to be propagated correctly by an algorithm.

Lemma 3. *Let the weights α, β such that $0 < \alpha < \beta \leq 2\alpha$, the ns B, and the point $\mathbf{p} \in X$ be given. Any fastest minimal cost (α,β)-weighted B-path with minimal number of 2-steps $\mathcal{P}_{\mathbf{q},\mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n = \mathbf{p}\rangle$ (for some $\mathbf{q} \in \overline{X}$ such that $d_{\alpha,\beta}\left(\mathbf{p}, \overline{X}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}}\right)$) satisfies* `(ii)` *in Definition 6, i.e.,*

$$\mathbf{p}_i, \mathbf{p}_{i+1} \textit{ are } b\left(DT_{\mathcal{L}}\left(\mathbf{p}_i\right) + 1\right) - \textit{neighbors for all } i.$$

Proof. Given a $\mathbf{p} \in X$, assume that there is a fastest minimal cost (α,β)-weighted B-path $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ with minimal number of 2-steps such that $\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}}\right) = d_{\alpha,\beta}(\mathbf{p}, \overline{X})$ and a $K < n$ such that $\mathcal{P}_{\mathbf{q},\mathbf{p}_K}$ is *not* represented by $DT_{\mathcal{L}}$, i.e., that $\mathbf{p}_{K-1}, \mathbf{p}_K$ are not $b(DT_{\mathcal{L}}(\mathbf{p}_{K-1}) + 1)$-neighbors. (Otherwise the value $\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_K}\right)$ is propagated from $DT_{\mathcal{C}}(\mathbf{p}_{K-1})$.) It follows that the values of $DT_{\mathcal{C}}(\mathbf{p}_{K-1})$ and $DT_{\mathcal{L}}(\mathbf{p}_{K-1})$ are given by a path $\mathcal{Q}_{\mathbf{q}',\mathbf{p}} = \langle \mathbf{q}' = \mathbf{p}'_0, \mathbf{p}'_1, \ldots, \mathbf{p}'_k = \mathbf{p}_i, \mathbf{p}'_{k+1}, \ldots, \mathbf{p}'_n = \mathbf{p}\rangle$ for some i and some $\mathbf{q}' \in \overline{X}$ such that

$$\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_{K-1}}\right) = d_{\alpha,\beta}\left(\mathbf{p}_{K-1}, \mathbf{q}; B\right) = d_{\alpha,\beta}\left(\mathbf{p}_{K-1}, \mathbf{q}'; B\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_{K-1}}\right)$$

and

$$\mathcal{L}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_{K-1}}\right) \neq \mathcal{L}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_{K-1}}\right).$$

We follow the cases in the proof of Lemma 2:

`Case i`: $\mathcal{L}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_{K-1}}\right) < \mathcal{L}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_{K-1}}\right)$

`Case i(a)`: $\mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_{K-1}}} > \mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_{K-1}}}$

We get $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}}\right)$ from the proof of Lemma 2 and $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_{K-1}}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_{K-1}}\right)$ by construction. Since $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ is a fastest path, $\mathbf{p}_{K-1}, \mathbf{p}_K$ is a 1-step, so it is also a $b(DT_{\mathcal{L}}(\mathbf{p}_{K-1}) + 1)$-step. Therefore, $\mathbf{p}_{K-1}, \mathbf{p}_K$ *are* $b(DT_{\mathcal{L}}(\mathbf{p}_{K-1}) + 1)$-neighbors. Contradiction.

`Case i(b)`: $\mathbf{2}_{\mathcal{Q}_{\mathbf{q}',\mathbf{p}_{K-1}}} \leq \mathbf{2}_{\mathcal{P}_{\mathbf{q},\mathbf{p}_{K-1}}}$

Following the proof of Lemma 2, we get $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}}\right) < \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}}\right)$.

`Case ii`: $\mathcal{L}\left(\mathcal{Q}_{\mathbf{q}',\mathbf{p}_{K-1}}\right) > \mathcal{L}\left(\mathcal{P}_{\mathbf{q},\mathbf{p}_{K-1}}\right)$

This leads to a longer B-path $\mathcal{Q}''_{\mathbf{q}',\mathbf{p}}$ with lower (or equal) cost by the construction in the proof of Lemma 2, so this case leads to a contradiction since $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ is a B-path of maximal length (see Remark 1) by assumption. □

3.1. Algorithms

In this section, algorithms for computing DTs using the additional transform $DT_{\mathcal{L}}$ are presented. First, we focus on a wavefront propagation algorithm. By Theorem 2, there is a distance propagating path for each $\mathbf{p} \in X$. This proves the correctness of Algorithm 1.

Algorithm 1: Computing $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$ for weighted ns-distances by wave-front propagation.

Input: B, α, β, neighborhoods $\mathcal{N}_1$ and $\mathcal{N}_2$, and an object $X \subset \mathbb{Z}^2$.
Output: The distance transforms $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$.
Initialization: Set $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$. Set $DT_{\mathcal{L}} = DT_{\mathcal{C}}$. For all grid points $\mathbf{p} \in \overline{X}$ adjacent to X: push $(\mathbf{p}, DT_{\mathcal{C}}(\mathbf{p}))$ to the list L of ordered pairs sorted by increasing $DT_{\mathcal{C}}(\mathbf{p})$.
Notation: $\omega_{\mathbf{v}}$ is α if $\mathbf{v} \in \mathcal{N}_1$ and β if $\mathbf{v} \in \mathcal{N}_2$.

```
while L is not empty do
    foreach p in L with smallest DT_C(p) do
        Pop (p, DT_C(p)) from L;
        foreach q: q, p are b(DT_L(p) + 1)-neighbors do
            if DT_C(q) > DT_C(p) + ω_{p−q} then
                DT_C(q) ← DT_C(p) + ω_{p−q};
                DT_L(q) ← DT_L(p) + 1;
                Push (q, DT_C(q)) to L;
            end
        end
    end
end
```

Now, the focus is on the raster-scanning algorithm. We will see that the DT can be computed correctly in three scans. Since a fixed number of scans is used and the time complexity is bounded by a constant for each visited grid point, the time complexity is linear in the number of grid points in the image domain.

We recall the following lemma from [14]:

Lemma 4. *When $\alpha < \beta \leq 2\alpha$, any minimal cost-path between $(0,0)$ and (x,y), where $x \geq y \geq 0$, consists only of the steps $(1,0)$, $(1,1)$, and $(0,1)$.*

Since we consider only "rectangular" image domains, the following lemma holds.

Lemma 5. *Given two points $\mathbf{p}, \mathbf{q}$ in $\mathcal{I}$, a ns B and weights $\beta > \alpha$. Any point in any minimal cost (α, β)-weighted B-path between $\mathbf{p}$ and $\mathbf{q}$ is in the image domain.*

Proof. Consider the point $\mathbf{p} = (x, y)$, where $x \geq y \geq 0$. By Lemma 4, *any* (α, β)-weighted B-path of minimal cost from $\mathbf{0}$ to $\mathbf{p}$ consists only of the local steps $(0,1), (1,1), (1,0)$. The theorem follows from this result. □

Let $\mathcal{N}^1 = \{(1,0),(1,1)\}, \mathcal{N}^2 = \{(1,1),(0,1)\}, \ldots, \mathcal{N}^8 = \{(1,-1),(1,0)\}$. In other words, the set $\mathcal{N}$ is divided into set according to which octant they belong.

Lemma 6. *Let γ be any permutation of $1, 2, \ldots, 8$. Between any two points, there is a distance propagating B-path $\mathcal{P}_{\mathbf{q},\mathbf{p}} = \langle \mathbf{q} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n = \mathbf{p}\rangle$ and integers $0 \leq K_1 \leq K_2 \leq \cdots \leq K_8 = n$ such that*

$$\begin{array}{lll}
\mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^{\gamma(1)} & \textit{if} & i \leq K_1 \\
\mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^{\gamma(2)} & \textit{if} & i > K_1 \textit{ and } i \leq K_2 \\
 & \vdots & \\
\mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^{\gamma(8)} & \textit{if} & i > K_7 \textit{ and } i \leq K_8.
\end{array}$$

Proof. Consider $\mathbf{q} = \mathbf{0}$ and $\mathbf{p} = (x, y)$ such that $x \geq y \geq 0$. Any minimal minimal cost B-path consists only of the local steps $(1,0), (1,1), (0,1)$ by Lemma 4. Reordering the 1-steps does not affect the cost of the path. The path obtained by reordering the 1-steps in a fastest minimal cost (α, β)-weighted B-path with minimal number of 2-steps is still a fastest minimal cost (α, β)-weighted B-path with minimal number of 2-steps. Therefore, there are distance propagating B-path such that

$$\begin{array}{lll}
\mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^1 & \text{if} & i \leq K_1 \\
\mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^2 & \text{if} & i > K_1 \text{ and } i \leq K_2
\end{array}$$

for some integers $0 \leq K_1 \leq K_2 = n$ and

$$\begin{array}{lll}
\mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^2 & \text{if} & i \leq K_1 \\
\mathbf{p}_i - \mathbf{p}_{i-1} \in \mathcal{N}^1 & \text{if} & i > K_1 \text{ and } i \leq K_2
\end{array}$$

for some integers $0 \leq K_1 \leq K_2 = n$. The general case follows from translation and rotation invariance. □

Definition 9. A *scanning mask* is a subset $\mathcal{M} \subset \mathcal{N}$.

Definition 10. A *scanning order (so)* is an enumeration of the $M = \text{card}(\mathcal{I})$ distinct points in $\mathcal{I}$, denoted $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M$.

Definition 11. Let $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M \in \mathcal{I}$ be a scanning order and $\mathcal{M}$ a scanning mask. The scanning mask $\mathcal{M}$ *supports the scanning order* if

$$\forall \mathbf{p}_i, \forall \mathbf{v} \in \mathcal{M}, \left((\exists i' > i : \mathbf{p}_{i'} = \mathbf{p}_i + \mathbf{v}) \text{ or } (\mathbf{p}_i + \mathbf{v} \notin \mathcal{I}_{\mathbb{G}}) \right).$$

Algorithm 2: Computing $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$ for weighted ns-distances by raster scanning.

Input: B, α, β, scanning masks $\mathcal{M}^i$, and an object $X \subset \mathbb{Z}^2$.
Output: The distance transforms $DT_{\mathcal{C}}$ and $DT_{\mathcal{L}}$.
Initialization: Set $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$. Set $DT_{\mathcal{L}} = DT_{\mathcal{C}}$.
Comment: The image domain $\mathcal{I}$ defined by eq. 1 is scanned L times using scanning orders such that the scanning mask $\mathcal{M}^i$ supports the scanning order so_i, $i \in \{1, \ldots, L\}$.
Notation: $\omega_{\mathbf{v}}$ is α if $\mathbf{v} \in \mathcal{N}_1$ and β if $\mathbf{v} \in \mathcal{N}_2$.

for $i = 1 : L$ **do**
  foreach $\mathbf{p} \in \mathcal{I}$ *following* so_i **do**
    if $DT_{\mathcal{C}}(\mathbf{p}) < \infty$ **then**
      foreach $\mathbf{v} \in \mathcal{M}^i$ **do**
        if $\mathbf{p}$ *and* $\mathbf{p} + \mathbf{v}$ *are* $b\left(DT_{\mathcal{L}}(\mathbf{p}) + 1\right)$*-neighbors* **then**
          if $DT_{\mathcal{C}}(\mathbf{p} + \mathbf{v}) > DT_{\mathcal{C}}(\mathbf{p}) + \omega_{\mathbf{v}}$ **then**
            $DT_{\mathcal{C}}(\mathbf{p} + \mathbf{v}) \leftarrow DT_{\mathcal{C}}(\mathbf{p}) + \omega_{\mathbf{v}}$;
            $DT_{\mathcal{L}}(\mathbf{p} + \mathbf{v}) \leftarrow DT_{\mathcal{L}}(\mathbf{p}) + 1$;
          end
        end
      end
    end
  end
end

Theorem 3. *If*

- *each of the sets* $\mathcal{N}^1, \mathcal{N}^2, \ldots, \mathcal{N}^8$ *is represented by at least one scanning mask and*
- *the scanning masks support the scanning orders,*

then Algorithm 2 computes correct distance maps.

Proof. Any distance propagating path between any pair of grid points in $\mathcal{I}$ is also in $\mathcal{I}$ by Lemma 5. Since the scanning masks support the scanning orders, there is, by Lemma 6, a distance propagating path that is propagated by the scanning masks. □

Corollary 1. *Algorithm 2 with, e.g., the masks*

$$\begin{aligned}\mathcal{M}^1 &= \{(-1,1),(-1,0),(-1,-1),(0,-1)\}\,,\\ \mathcal{M}^2 &= \{(0,-1),(1,-1),(1,0),(1,1)\}\,,\ \textit{and}\\ \mathcal{M}^3 &= \{(-1,1),(0,1),(1,1)\}\,,\end{aligned}$$

see Figure 4, gives correct distance transforms.

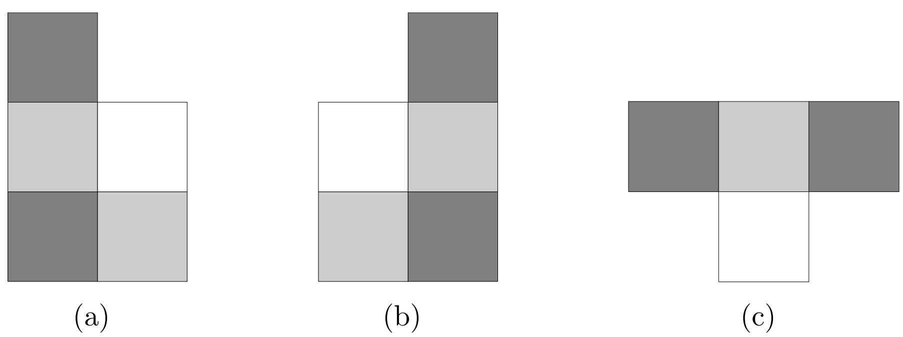


Figure 4: Masks that can be used with Algorithm 2. The white pixel is the center of the mask and the two grey levels correspond to the elements in $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively.

4. Computing the distance transform using a look-up table

The look-up table $LUT_{\mathbf{v}}(k)$ gives the value to be propagated in the direction $\mathbf{v}$ from a grid point with distance value k. We will see that by using this approach, the additional distance transform $DT_{\mathcal{L}}$ is *not* needed for computing $DT_{\mathcal{C}}$. Thus, we get an efficient algorithm in this way. In this section, we assume that integer weights are used.

The LUT-based approach to compute the distance transform first appeared in [15]. The distance function considered in [15] uses neighborhood sequences, but is non-symmetric. The non-symmetry allows to compute the DT in one scan. The same LUT-based approach is used for binary mathematical morphology with convex structuring elements in [16]. This approach is efficient for, e.g., binary erosion in one scan with a computational per-pixel cost independent of the size of the structuring element.

For the algorithm in [15, 16] the following formula is used in one raster scan:

$$DT_{\mathcal{C}}(\mathbf{p}) = \min_{\mathbf{v}\in\mathcal{N}} \left(LUT_{\mathbf{v}} \left(DT_{\mathcal{C}} (\mathbf{p}+\mathbf{v}) \right) \right),$$

where $\mathcal{N}$ is a non-symmetric neighborhood. In this section, we will extend this approach and allow the (symmetric) weighted ns-distances by allowing more than one scan.

4.1. Construction of the look-up table

Given a distance value k, the look-up table at position k with subscript-vector $\mathbf{v}$, $LUT_{\mathbf{v}}(k)$, holds information about the *maximal* distance value that can be found in a distance map in direction $\mathbf{v}$. See Example 3 and 4.

Example 3. For a distance function on $\mathbb{Z}^2$ defined by $\alpha = 2$, $\beta = 3$ and $B = (1, 2)$, the LUT with $D_{\max} = 10$ is the following:

	j	$\underline{0}$	1	$\underline{2}$	3	$\underline{4}$	$\underline{5}$	$\underline{6}$	$\underline{7}$	$\underline{8}$	$\underline{9}$	$\underline{10}$
$\mathbf{v} \in \mathcal{N}_1$	$LUT_{\mathbf{v}}(j)$	2	3	4	5	6	7	8	9	10	11	12
$\mathbf{v} \in \mathcal{N}_2$	$LUT_{\mathbf{v}}(j)$	4	4	5	6	7	9	9	10	11	12	14

Only the values that are underlined are attained by the distance functions. See also Figure 5. The values in the look-up tables can be extracted from these DTs by, for each distance value 0 to 10, finding the corresponding maximal value in the subscript-direction.

Example 4. For a distance function on $\mathbb{Z}^2$ defined by $\alpha = 4$, $\beta = 5$ and $B = (1, 2, 1, 2, 2)$, the LUT (only showing values that are attained by the distance function) with $D_{\max} = 23$ is the following:

	j	0	4	8	9	12	13	16	17	18	20	21	22	23
$\mathbf{v} \in \mathcal{N}_1$	$LUT_{\mathbf{v}}(j)$	4	8	12	13	16	17	20	21	22	24	25	26	27
$\mathbf{v} \in \mathcal{N}_2$	$LUT_{\mathbf{v}}(j)$	8	9	13	17	17	18	21	22	23	25	26	27	31

The values in the LUT are given by the formula in Lemma 7.

14							
12	13	14					
10	11	12	14				
8	9	10	12	14			
6	7	9	10	12	14		
4	5	7	9	10	12	14	
2	4	5	7	9	11	13	
0	2	4	6	8	10	12	14

Figure 5: Each pixel above is labeled with the distance to the pixel with value 0. The parameters $B = (1, 2)$, $(\alpha, \beta) = (2, 3)$ are used. See also Example 3.

Lemma 7. *Let α, β such that $0 < \alpha \leq \beta \leq 2\alpha$, the ns B, and the integer value k be given. Then*

$$\max_{\left\{\substack{\mathbf{v}\in\mathcal{N}_1 \\ \mathbf{p}:\ d_{\alpha,\beta}(\mathbf{0},\mathbf{p};B)=k}\right\}} \left(d_{\alpha,\beta}\left(\mathbf{0}, \mathbf{p}+\mathbf{v}; B\right) - k\right) = \alpha \textit{ and}$$

$$\max_{\left\{\substack{\mathbf{v}\in\mathcal{N}_2 \\ \mathbf{p}:\ d_{\alpha,\beta}(\mathbf{0},\mathbf{p};B)=k}\right\}} \left(d_{\alpha,\beta}\left(\mathbf{0}, \mathbf{p}+\mathbf{v}; B\right) - k\right) = \begin{cases} 2\alpha & \textit{if } \exists n: \ b(n+1) = 1 \\ & \textit{and } k = \mathbf{1}_B^n\alpha + \mathbf{2}_B^n\beta \\ \beta & \textit{else.} \end{cases}$$

Proof. When $\mathbf{v}$ is a 1-step, then the maximum difference between $d_{\alpha,\beta}\left(\mathbf{0}, \mathbf{p}+\mathbf{v}; B\right)$ and $d_{\alpha,\beta}\left(\mathbf{0}, \mathbf{p}; B\right)$ is α by definition. There is a local step $\mathbf{v} \in \mathcal{N}_1$ that increases the length of the minimal cost B-path (for any B) by 1, so the maximum difference α is always attained.

When $\mathbf{v}$ is a strict 2-step, $\mathbf{v} \in \mathcal{N}_2$ is the sum of two local steps from $\mathcal{N}_1$. Intuitively, if there are "enough" 2s in B, then the maximum difference is β. Otherwise, two 1-steps are used and the maximum difference is 2α. To prove this, let $\mathcal{P}_{\mathbf{0},\mathbf{p}} = \langle \mathbf{0} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n = \mathbf{p} \rangle$ be a minimal cost B-path and let $\mathbf{p} = (x, y)$ be such that $x \geq y \geq 0$. We have the following conditions on B:

(i) $b(n+1) = 1$ and

(ii) $d_{\alpha,\beta}(\mathbf{0}, \mathbf{p}; B) = \mathbf{1}_B^n \alpha + \mathbf{2}_B^n \beta$.

We note that (i) implies that $\mathcal{P}_{\mathbf{0},\mathbf{p}} \cdot \langle \mathbf{p} + \mathbf{w} \rangle$ is a B-path iff $\mathbf{w} \in \mathcal{N}_1$ and a minimal cost B-path if $\mathbf{w}$ is either $(1,0)$ or $(0,1)$. Also, (ii) implies that the number of 2:s in B up to position n equals the number 2-steps in $\mathcal{P}_{\mathbf{0},\mathbf{p}}$.

If both (i) *and* (ii) are fulfilled, since $b(n+1) = 1$, the 2-step $\mathbf{v} = (1,1)$ is divided into two 1-steps $(1,0)$ and $(0,1)$ giving a minimal cost B-path, so $d_{\alpha,\beta}(\mathbf{0}, \mathbf{p} + \mathbf{v}; B) = d_{\alpha,\beta}(\mathbf{0}, \mathbf{p}; B) + 2\alpha$.

If (i) is not fulfilled, then there is a 2-step $\mathbf{v}$ such that $\mathcal{P}_{\mathbf{0},\mathbf{p}} \cdot \langle \mathbf{p} + \mathbf{v} \rangle$ is a minimal cost B-path of cost $d_{\alpha,\beta}(\mathbf{0}, \mathbf{p} + \mathbf{v}; B) = d_{\alpha,\beta}(\mathbf{0}, \mathbf{p}; B) + \beta$.

If (i), but not (ii) is fulfilled, then for any minimal cost B-path $\mathcal{Q}_{\mathbf{0},\mathbf{p}}$, we have

$$k = \mathbf{1}_{\mathcal{Q}_{\mathbf{0},\mathbf{p}}} \alpha + \mathbf{2}_{\mathcal{Q}_{\mathbf{0},\mathbf{p}}} \beta \neq \mathbf{1}_B^{\mathcal{L}(\mathcal{Q}_{\mathbf{0},\mathbf{p}})} \alpha + \mathbf{2}_B^{\mathcal{L}(\mathcal{Q}_{\mathbf{0},\mathbf{p}})} \beta.$$

It follows that $\mathbf{2}_{\mathcal{Q}_{\mathbf{0},\mathbf{p}}} < \mathbf{2}_B^{\mathcal{L}(\mathcal{Q}_{\mathbf{0},\mathbf{p}})}$. Therefore, there is a 1-step in $\mathcal{Q}_{\mathbf{0},\mathbf{p}}$ that can be swapped with the 2-step $\mathbf{v}$ giving a minimal cost B-path of cost $d_{\alpha,\beta}(\mathbf{0}, \mathbf{p}; B) + \beta$. $\square$

The formula in Lemma 7 gives an efficient way to compute the look-up table, see Algorithm 3. The algorithm gives a correct LUT by Lemma 7. The output of Algorithm 3 for some parameters is shown in Example 3 and 4.

```
Algorithm 3: Computing the look-up table for weighted ns-distances.
Input: Neighborhoods N_1, N_2, weights α and β (0 < α ≤ β ≤ 2α), a
       ns B, and the largest distance value D_max.
Output: The look-up table LUT.
for k = 1 : D_max do
    foreach v ∈ N_1 do
        LUT_v(k) ← k + α;
    end
    foreach v ∈ N_2 do
        LUT_v(k) ← k + β;
    end
end
n ← 0;
while 1_B^n α + 2_B^n β ≤ D_max do
    if b(n + 1) == 1 then
        LUT_v(1_B^n α + 2_B^n β) ← (1_B^n + 2) α + 2_B^n β;
    end
    n ← n + 1;
end
```

Lemma 8 shows that the distance values are propagated correctly along distance propagating paths by using the look-up table.

Lemma 8. *Let $\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_n} = \langle \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n \rangle$ be a distance propagating B-path. Then*

$$\mathcal{C}_{\alpha,\beta}\left(\langle \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{i+1} \rangle\right) = LUT_{\mathbf{p}_{i+1}-\mathbf{p}_i}\left(\mathcal{C}_{\alpha,\beta}\left(\langle \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_i \rangle\right)\right) \forall i < n.$$

Proof. Assume that the lemma is false and let i be the minimal index such that

$$\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}}\right) \neq LUT_{\mathbf{p}_{i+1}-\mathbf{p}_i}\left(\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right)\right).$$

Then there is a path $\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j} = \langle \mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_j \rangle$ such that

$$\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right) = \mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}\right) \quad \text{and} \quad \mathcal{L}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right) \neq \mathcal{L}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}\right)$$

defining the value in the LUT, i.e.,

$$\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j} \cdot \langle \mathbf{q}_j + \mathbf{v} \rangle\right) = LUT_{\mathbf{v}}\left(\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right)\right),$$

where $\mathbf{v} = \mathbf{p}_{i+1} - \mathbf{p}_i$. Since the LUT stores the maximal local distances that are attained,

$$\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j} \cdot \langle \mathbf{q}_j + \mathbf{v} \rangle\right) > \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}}\right).$$

It follows from Lemma 7 that $\mathbf{v}$ is a strict 2-step and that $LUT_{\mathbf{v}}\left(\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right)\right) = 2\alpha$ and

$$\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}}\right) - \mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right) = \beta.$$

Since $\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}$ is a distance propagating B-path and $\mathbf{v}$ is a strict 2-step,

$$\mathbf{2}_B^{\mathcal{L}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right)} = \mathbf{2}_{\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}} \tag{3}$$

and $\mathbf{2}_B^{\mathcal{L}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}}\right)} = \mathbf{2}_{\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_{i+1}}}$.

`case i` $\mathcal{L}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right) > \mathcal{L}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}\right)$

It follows that $\mathbf{2}_{\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}} < \mathbf{2}_{\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}} \leq \mathbf{2}_B^{\mathcal{L}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}\right)} \leq \mathbf{2}_B^{\mathcal{L}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right)}$ which contradicts (3).

`case ii` $\mathcal{L}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right) < \mathcal{L}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}\right)$

This implies that $\mathbf{2}_{\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}} > \mathbf{2}_{\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}}$. Then $\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j} \cdot \langle \mathbf{q}_j + \mathbf{v} \rangle$ is not a distance propagating path (there are more elements 2 in B than 2-steps in the path). It follows from Lemma 7 that there is a distance propagating path from $\mathbf{q}_0$ to $\mathbf{q}_j + \mathbf{v}$ of cost $\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}\right) + \beta$. Since $\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}$ is arbitrary, it follows that $LUT_{\mathbf{v}}\left(\mathcal{C}_{\alpha,\beta}\left(\mathcal{P}_{\mathbf{p}_0,\mathbf{p}_i}\right)\right) = LUT_{\mathbf{v}}\left(\mathcal{C}_{\alpha,\beta}\left(\mathcal{Q}_{\mathbf{q}_0,\mathbf{q}_j}\right)\right) = \beta$. Contradiction. □

4.2. Algorithms for computing the DT using look-up tables

In this section, we give algorithms that can be used to compute the distance transform using the LUT-approach. By Lemma 8, distance values are propagated correctly along distance propagating paths, so Algorithm 4 produces correct distance maps.

Algorithm 4: Computing $DT_{\mathcal{C}}$ for weighted ns-distances by wave-front propagation using a look-up table.

Input: LUT and an object $X \subset \mathbb{Z}^2$.

Output: The distance transform $DT_{\mathcal{C}}$.

Initialization: Set $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$. For all grid points $\mathbf{p} \in \overline{X}$ adjacent to X: push $(\mathbf{p}, DT_{\mathcal{C}}(\mathbf{p}))$ to the list L of ordered pairs sorted by increasing $DT_{\mathcal{C}}(\mathbf{p})$.

```
while L is not empty do
    foreach p in L with smallest DT_C(p) do
        Pop (p, DT_C(p)) from L;
        foreach v ∈ N do
            if DT_C(p + v) > LUT_v (DT_C(p)) then
                DT_C(p + v) ← LUT_v (DT_C(p));
                Push (p + v, DT_C(p + v)) to L;
            end
        end
    end
end
```

Theorem 4. *If*

- *each of the sets* $\mathcal{N}^1, \mathcal{N}^2, \ldots, \mathcal{N}^8$ *is represented by at least one scanning mask and*
- *the scanning masks support the scanning orders,*

then Algorithm 5 computes correct distance maps.

Proof. Since the same paths are propagated using this technique, the same conditions on the masks, scanning orders, and image domain are needed for Algorithm 5 to produce distance transforms without errors as when the additional distance transform $DT_{\mathcal{L}}$ is used. □

Note that Algorithm 4 and 5 derive from the work in [15, 16], but here, symmetrical distance functions are allowed due to the increased number of scans.

Algorithm 5: Computing $DT_{\mathcal{C}}$ for weighted ns-distances by raster scanning using a look-up table.

Input: LUT, scanning masks $\mathcal{M}^i$, scanning orders so_i and an object $X \subset \mathbb{Z}^2$.
Output: The distance transform $DT_{\mathcal{C}}$.
Initialization: Set $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$.
Comment: The image domain $\mathcal{I}$ defined by eq. 1 is scanned L times using scanning orders such that the scanning masks $\mathcal{M}^i$ supports the scanning order so_i, $i \in \{1, \ldots, L\}$

```
for i = 1 : L do
    foreach p ∈ I following so_i do
        if DT_C(p) < ∞ then
            foreach v ∈ N^i do
                if DT_C(p + v) > LUT_v (DT_C(p)) then
                    DT_C(p + v) ← LUT_v (DT_C(p));
                end
            end
        end
    end
end
```

We remark that the computational cost of Algorithm 3 is linear with respect to the maximal radius $D_{\max}$ and that the LUT can be computed on the fly when computing the DT. In other words, if it turns out during the DT computation that the LUT is too short, it can be extended by using Algorithm 3 with the modification that the loop variable starts from the missing value. Note also that for short neighborhood sequences, the LUT can sometimes be replaced by a modulo operator. For example, when $(\alpha, \beta) = (4, 5)$ and $B = (1, 2)$, then by propagating distances to 2-neighbors only when $DT_{\mathcal{C}}(\mathbf{p})$ is not divisible by nine gives a very fast algorithm the computes a DT with low rotational dependency.

5. Computing the distance transform in two scans using a large mask

In [6], it is proved that if the weights and neighborhood sequence are such that the generated distance function is a metric, then the distance function is generated by constant neighborhood using a (large) neighborhood. This implies that the 2-scan chamfer algorithm can be used to compute the DT in $\mathbb{Z}^2$, see [12].

The following theorem is proved in [11]:

Theorem 5. *If*

$$\sum_{i=1}^{N} b(i) \leq \sum_{i=j}^{j+N-1} b(i) \quad \forall j, N \geq 1 \quad \textit{and} \tag{4}$$

$$0 < \alpha \leq \beta \leq 2\alpha \tag{5}$$

then $d_{\alpha,\beta}(\cdot,\cdot;B)$ *is a metric.*

In [11], the distance function generated by $B = (1,2,1,2,2)$, $(\alpha,\beta) = (4,5)$ is suggested. In Figure 6, the masks that can be used by a two-scan algorithm to compute the DT with this distance function are shown.

						4	8	12	16	20
21	17	13	9	8	4	8	9	13	17	21
22	18	17	13	9	8	9	13	17	18	22
23	22	18	17	13	12	13	17	18	22	23
	23	22	18	17	16	17	18	22	23	
		23	22	21	20	21	22	23		

(a)

		23	22	21	20	21	22	23		
	23	22	18	17	16	17	18	22	23	
23	22	18	17	13	12	13	17	18	22	23
22	18	17	13	9	8	9	13	17	18	22
21	17	13	9	8	4	8	9	13	17	21
20	16	12	8	4						

(b)

Figure 6: Masks that can be used by a two-scan algorithm to compute a DT using the weighted ns-distance defined by $B = (1,2,1,2,2)$, $(\alpha,\beta) = (4,5)$.

In this section we assume that the weights α and β and the ns B are such that

- B is periodic and
- the distance function generated by α, β, and B is a metric.

For this family of distance functions, a two-scan chamfer algorithm with large scanning masks can be used instead of the three-scan algorithm with small scanning masks using DT_{length} or the three-scan algorithm with small scanning mask using a LUT.

Let $\overline{\mathcal{N}}$ be the set of grid points such that the distance value from $\mathbf{0}$ is defined by the first period of B.

We now define two sets that are used in Algorithm 6.

$$\begin{aligned}
\mathcal{M}^1 &= \overline{\mathcal{N}} \cap \{(x,y) : y < 0 \text{ or } y = 0 \text{ and } x \geq 0\} \text{ and} \\
\mathcal{M}^2 &= \overline{\mathcal{N}} \cap \{(x,y) : y > 0 \text{ or } y = 0 \text{ and } x \leq 0\}.
\end{aligned}$$

The following theorem is proved in [6]:

Theorem 6. *If $d_{\alpha,\beta}(\cdot,\cdot;B)$ is a metric, then the weighted ns-distance defined by B and (α,β) defines the same distance function as the weighted distance defined by the weighted vectors*

$$\left\{ \Big(\mathbf{v}, d_{\alpha,\beta}(\mathbf{0}, \mathbf{0}+\mathbf{v}; B) \Big) : \mathbf{v} \in \overline{\mathcal{N}} \right\}$$

Algorithm 6: Computing $DT_{\mathcal{C}}$ for weighted ns-distances by wave-front propagation using a large weighted mask.

Input: The mask $\overline{\mathcal{N}}$, and an object $X \subset \mathbb{Z}^2$.

Output: The distance transform $DT_{\mathcal{C}}$.

Initialization: Set $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow 0$ for grid points $\mathbf{p} \in \overline{X}$ and $DT_{\mathcal{C}}(\mathbf{p}) \leftarrow \infty$ for grid points $\mathbf{p} \in X$. For all grid points $\mathbf{p} \in \overline{X}$ adjacent to X: push $(\mathbf{p}, DT_{\mathcal{C}}(\mathbf{p}))$ to the list L of ordered pairs sorted by increasing $DT_{\mathcal{C}}(\mathbf{p})$.

```
while L is not empty do
    foreach p in L with smallest DT_C(p) do
        Pop (p, DT_C(p)) from L;
        foreach v ∈ N̄ do
            if DT_C(p + v) > DT_C(p) + ω_v then
                DT_C(p + v) ← DT_C(p) + ω_v;
                Push (p + v, DT_C(p + v)) to L;
            end
        end
    end
end
```

Algorithm 7: Computing $DT_{\mathcal{C}}$ for weighted ns-distances by two scans using a large weighted mask.

```
Input: Scanning masks 𝓜^i, scanning orders so_i, weights, and an object X ⊂ ℤ².
Output: The distance transform DT_C.
Initialization: Set DT_C(p) ← 0 for grid points p ∈ X̄ and
DT_C(p) ← ∞ for grid points p ∈ X.
Comment: The image domain 𝓘 defined by eq. 1 is scanned two
times using scanning orders such that the scanning mask defined by
𝓜^i supports the scanning order so_i, i ∈ {1, ..., 2}
for i = 1 : 2 do
    foreach p ∈ 𝓘 following so_i do
        foreach v ∈ 𝓜^i do
            if DT_C(p + v) > DT_C(p) + ω_v then
                DT_C(p + v) ← DT_C(p) + ω_v;
            end
        end
    end
end
```

Theorem 7. *If the scanning masks $\mathcal{M}^1$ and $\mathcal{M}^2$ support the scanning orders then Algorithm 6 and 7 compute correct distance maps.*

Proof. Any path consists of steps from $\overline{\mathcal{N}}$ and the order of the steps is arbitrary. Consider the point $\mathbf{p} = (x, y)$ such that $x \geq y \geq 0$. All points in any minimal cost path between $\mathbf{0}$ and $\mathbf{p}$ have non-negative coordinates. Also, all local steps are in $\mathcal{M}^2$ except $(1, 0)$. Thus, the local steps in any minimal cost path between $\mathbf{0}$ and $\mathbf{p}$ can be rearranged such that the steps from $\mathcal{M}^1$ are first and the the steps from $\mathcal{M}^2$ are last or vice-versa. It follows that a minimal cost path is propagated from $\mathbf{0}$ to each point $\mathbf{p}$ such that $x \geq y \geq 0$. The theorem holds by translation and rotation invariance. $\square$

6. Conclusions

We have examined the DT computation for weighted ns-distances. Three different, but related, algorithms have been presented and we have proved that the resulting DTs are correct.

We have shown that using the additional transform $DT_{\mathcal{L}}$ is not needed for computing the DT $DT_{\mathcal{C}}$. This extra information can, however, be useful when extracting medial representations, see [14].

We note that when the LUT-approach is used, a fast and efficient algorithm is obtained. This approach can also be used for computing the *constrained* DT. When the constrained DT is computed, there are obstacle grid points that are not allowed to intersect with the minimal cost paths that define the distance values. The path-based approach is well-suited for such algorithms. When the Euclidean distance is used, the corresponding algorithm must keep track of *visible* point, i.e., points which can be given the distance value by adding the length of the straight line segment between already visited points. Such algorithms, see [17], are slow and computationally heavy compared to the distance functions used in this paper.

For short sequences, the LUT can be replaced by a modulo function: consider $B = (1, 2)$ and weights (α, β), then β is propagated to a two neighbor only from grid points with distance values that are divisible by $\alpha + \beta$. This approach gives a fast and efficient algorithm.

The LUT can be computed "on-the-fly" by using Algorithm 3. In other words, if it turns out during the DT computation that the LUT is too short, Algorithm 3 can be used to find the missing values in time that is proportional to the number of added values.

For long sequences, the two-scan Algorithm 6 and 7 is not efficient since the size of the masks depend on the length of the sequence. Also, this approach is valid only for metric distance functions.

Due to the low rotational dependency and the efficient algorithms presented here, we expect that the weighted ns-distance has the potential of being used in several image processing-applications where the DT is used: matching [18], morphology [19], and more recent applications such as separating arteries and veins in 3-D pulmonary CT, [20] and traffic sign recognition, [21].

References

[1] A. Rosenfeld, J. L. Pfaltz, Sequential operations in digital picture processing, Journal of the ACM 13 (4) (1966) 471–494.

[2] U. Montanari, A method for obtaining skeletons using a quasi-Euclidean distance, Journal of the ACM 15 (4) (1968) 600–624.

[3] G. Borgefors, Distance transformations in digital images, Computer Vision, Graphics, and Image Processing 34 (1986) 344–371.

[4] G. Borgefors, On digital distance transforms in three dimensions, Computer Vision and Image Understanding 64 (3) (1996) 368–376.

[5] A. Rosenfeld, J. L. Pfaltz, Distance functions on digital pictures, Pattern Recognition 1 (1968) 33–61.

[6] M. Yamashita, T. Ibaraki, Distances defined by neighbourhood sequences, Pattern Recognition 19 (3) (1986) 237–246.

[7] B. Nagy, Metric and non-metric distances on $\mathbb{Z}^n$ by generalized neighbourhood sequences, in: Proceedings of 4^{th} International Symposium on Image and Signal Processing and Analysis (ISPA 2005), Zagreb, Croatia, 2005, pp. 215–220.

[8] B. Nagy, Distances with neighbourhood sequences in cubic and triangular grids, Pattern Recognition Letters 28 (1) (2007) 99–109.

[9] C. Fouard, R. Strand, G. Borgefors, Weighted distance transforms generalized to modules and their computation on point lattices, Pattern Recognition 40 (9) (2007) 2453–2474.

[10] R. Strand, B. Nagy, C. Fouard, G. Borgefors, Generating distance maps with neighbourhood sequences, in: Proceedings of 13^{th} International Conference on Discrete Geometry for Computer Imagery (DGCI 2006), Szeged, Hungary, Vol. 4245 of Lecture Notes in Computer Science, Springer-Verlag, 2006, pp. 295–307.

[11] R. Strand, Weighted distances based on neighbourhood sequences, Pattern Recognition Letters 28 (15) (2007) 2029–2036.

[12] A. Hajdu, L. Hajdu, R. Tijdeman, General neighborhood sequences in $\mathbb{Z}^n$, Discrete Applied Mathematics 155 (2007) 2507–2522.

[13] R. Strand, B. Nagy, G. Borgefors, Digital distance functions on three-dimensional grids, Theoretical Computer Science 412 (15) (2011) 1350 – 1363, theoretical Computer Science Issues in Image Analysis and Processing.

[14] R. Strand, Shape representation with maximal path-points for path-based distances, in: Proceedings of 5^{th} International Symposium on Image and Signal Processing and Analysis (ISPA 2007), Istanbul, Turkey, 2007, pp. 397–402.

[15] N. Normand, C. Viard-Gaudin, A background based adaptive page segmentation algorithm, in: Proceedings of the 3^{rd} International Conference on Document Analysis and Recognition (ICDAR 1995), Washington, DC, USA, Vol. 1, 1995, pp. 138–141.

[16] N. Normand, Convex structuring element decomposition for single scan binary mathematical morphology, in: Proceedings of 11^{th} International Conference on Discrete Geometry for Computer Imagery (DGCI 2003), Naples, Italy, Vol. 2886 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 154–163.

[17] D. Coeurjolly, S. Miguet, L. Tougne, 2D and 3D visibility in discrete geometry: an application to discrete geodesic paths, Pattern Recognition Letters 25 (5) (2004) 561–570.

[18] G. Borgefors, Hierarchical chamfer matching: A parametric edge matching algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (6) (1988) 849–865.

[19] P. Nacken, Chamfer metrics in mathematical morphology, Journal of Mathematical Imaging and Vision 4 (1994) 233–253.

[20] P. Saha, Z. Gao, S. Alford, M. Sonka, E. Hoffman, Topomorphologic separation of fused isointensity objects via multiscale opening: Separating arteries and veins in 3-D pulmonary CT, IEEE Transactions on Medical Imaging 29 (3) (2010) 840 –851.

[21] A. Ruta, Y. Li, X. Liu, Real-time traffic sign recognition from video by class-specific discriminative features, Pattern Recognition 43 (1) (2010) 416 – 430.